Action Recognition using Rank-1 Approximation of Joint Self-Similarity Volume

Anonymous ICCV submission

Paper ID 587

Abstract

In this paper, we introduce a generic method for action recognition that does not depend on one particular type of input feature vector. We make three main contributions: (i) We introduce the concept of Joint Self-Similarity Volume (Joint SSV) for modeling dynamical systems, and show that by using a new optimized rank-1 tensor approximation of Joint SSV one can obtain compact low-dimensional descriptors that very accurately preserve the dynamics of the original system, e.g. an action video sequence; (ii) The descriptor vectors derived from the optimized rank-1 approximation make it possible to recognize actions without explicitly aligning the videos in time in order to compensate for speed of execution or differences in video frame rates; (iii) The method is generic and can be applied using different low-level features such as silhouettes, tracked points, histogram of oriented gradients, etc. Hence, it does not necessarily require explicit tracking of features in the space-time volume. Our experimental results on three public datasets demonstrate that our method produces remarkably good results and outperforms all baseline methods.

1. Introduction

Action recognition has continued to be an active area of research and has thus rightfully attracted much attention from the researchers over the years. Recent application domains, such as automatic video indexing and archiving, video surveillance, human-computer interactions, gesture recognition etc., will benefit immensely from a robust and an efficient solution to this problem. There are many factors that make this a challenging problem: large variations in performing an action by different people, whether by varying the postures, or the execution speed. Illumination variations in the sequences, occlusions and disocclusions, distracting background motions, and perspective effects and camera motions are some of the other factors that further complicate this problem. As a consequence, current methods often resort to restricted and simplified scenarios with simple backgrounds, simpler kinematic action classes, static cameras or limited view variations.

Various approaches have been proposed over the years for action recognition. On the basis of representation, they can be categorized as: Time evolution of human silhouettes [27], action cylinders, space-time shapes [29, 11, 12], local 3D patch analysis [17, 7, 21, 9], generally coupled with some machine learning techniques. Almost all the works mentioned above rely primarily on an effective feature extraction technique. These feature extraction methods can be roughly categorized into: motion-based [8], appearance based [12], space-time volume based [29, 11, 12], and space-time interest points or local features based [7, 18, 20, 9]. Motion-based methods generally compute optical flow from a given action sequence, followed by appropriate feature extraction. However, optical flow based methods are known to be very susceptible to noise and easily lead to inaccuracies. Appearance based methods are prone to differences in appearance between the training dataset and the test sequences. Volume or shape based methods require highly detailed silhouette extraction, which may not be possible in given real-world noisy video dataset. In comparison with these approaches, the space-time interest point (STIP) based methods [7, 20] are more robust to noise and camera movement and also seem to work quite well with low resolution inputs. However, these methods rely solely on the discriminative power of individual local space-time descriptors. Information related to the global spatio-temporal distribution is ignored. Thus due to lack of this temporal information, smooth motions cannot be captured using STIP methods. In addition, issues like optimal space-time descriptor selection and codebook clustering algorithm selection have to be addressed, with fine-tuning various parameters, which is highly data dependent [3].

The notion of “self-similarity” has received significant attention, recently. The work in [1] describes a gait recognition technique based on the image self-similarity of a walking person and classify the movement patterns of different people. Some works [5, 4] also show the effective use of the self-similarity in recognizing different types of biological periodic motions. The method that is closely related to ours is that of Junejo et al.[13]. The authors exploit the no-
tion of image self-similarities, as proposed by [24]. For a given action sequence, [13] first extract some low level features. The distances between extracted features for all pairs of time frames are computed and this results in a SSM. Each action sequence is thus reduced to a 2D SSM matrix, and the authors then proceed to extracting some useful features from these SSMs and use it to train the action recognition system.

In this work we address action recognition from a different perspective. In contrast to [1, 5, 4, 13] where they reduce a sequence to a 2D SSM, our formulation constructs a 3-order tensor for each action sequence, which we refer to as the Joint Self-Similarity Volume (Joint SSV). This is then followed by an optimized rank-1 tensor approximation.

This novel usage of the rank-1 approximation, first, avoids the need to pre-align videos; second, rank-1 tensor approximation gives a huge dimensionality reduction, and hence a significant saving in memory and computational time can be achieved because the reference database is just a collection of rank-1 tensors, and finally, we only need one rank-1 tensor per action in our reference database and require no training.

We evaluate three different schemes of constructing the Joint SSV, namely the HOG3D-based, the silhouette-based, and the tracked point based schemes, on three public datasets. The results are remarkably good, and outperform all baseline methods in our experiments.

This paper is organized as follows: Section 2 gives an overview of our method. Section 3 presents some preliminaries of SSM and Joint SSM. Sections 4 is dedicated to the construction of Joint SSV, followed by an optimized rank-1 tensor decomposition algorithm in Section 5. In Section 6, we provide a similarity measure for the decomposed vectors for their classification. All experiment results and analysis are presented in Section 6 and Section 7.

2. Overview of the method

Our framework is shown schematically in Fig. 1. We construct a SSM for each frame of the video sequence using a feature vector. We then extract Joint SSMs from this sequence of SSMs, leading to a Joint Self-Similarity Volume (Joint SSV). Joint SSV is then decomposed into its rank-1 approximation vectors using an optimized iterative tensor decomposition algorithm. This yields a set of compact vector descriptors that are highly discriminative between different actions. To evaluate our method on human action recognition, we used three public datasets. To show that our method is generic and does not depend on the input feature vector, we tested our method using various low-level features like silhouette and tracked points, as well as middle-level features like HOG3D. The final step used a nearest neighbor classification using the descriptor vectors produced by the rank-1 decomposition of Joint SSV.

Below, we first introduce some basic results on Joint SSM, an then describe how Joint SSV is constructed from a sequence of SSMs.

3. Joint Self-similarity Matrix

Here are some preliminary results on SSM:

Definition 1: An SSM can be expressed by a $N \times N$ matrix $R_{ij}(v, w) = \Theta(\epsilon - \|v_i - w_j\|_p)$, for $i, j = 1, ..., N$, where $N$ is the length of a feature vector $v$, and $\epsilon$ is a threshold distance.

Theorem 1: Given a threshold $\epsilon$ and a distance metric $d$, vector $v \in \mathbb{R}^d$ uniquely corresponds to a self-similarity matrix $M_v^\epsilon$.

Theorem 2: Let $M_v(u) = R_{ij}(v, u)$, for $p = 1$ and $\epsilon = 0$, we have $M_v(u \pm \epsilon) = |M_v(u) \pm M_v(\epsilon)|$.

It can be verified that the SSM holds the following properties: $R_{ij} = R_{ji}$ (symmetry); $R_{ij} \geq 0$ for all $i$ and $j$ (Positivity); and $R_{ik} \leq R_{ij} + R_{jk}$ for all $i, j, k$ (Triangle inequality), and hence it is a metric. SSM provides important insights into the dynamics of a vector, which is especially advantageous in high dimensional spaces [1, 5].

The intuition behind the SSM is that, according to recurrent plot theory, if we view the vector $v$ as a trajectory in 2D space, the SSM itself captures the internal dynamics of this trajectory in a matrix form [2].

We further extend the SSM to Joint SSM based on the idea of Joint Recurrence Plot (JPR) theory, and it will be used in our volume construction procedure.

Definition 2: The Joint SSM is defined as $JR_{ij}^v(u, \epsilon_v, \epsilon_w, v, w) = \Theta(\epsilon_v - \|v_i - w_j\|_p)\Theta(\epsilon_w - \|w_i - v_j\|_p)$, for $i, j = 1, ..., N$, $\epsilon_v$ and $\epsilon_w$ are two internal thresholds, $p_1$ and $p_2$ are two distance norms.

The motivation for this extension is that, $JR_{ij}^v$ can be viewed as defining the relationship between two trajectories, and represent their interaction in a uniform manner. In other words, a recurrence will take place if a point $v_i$ on the first trajectory $v$ returns to the neighborhood of a former point $v_j$, and simultaneously a point $w_j$ on the second trajectory $w$ returns to the neighborhood of a former point $w_i$. For this reason, this extension is advantageous when we want to fuse two SSMs, whose resulting SSM specifically encodes the mutual dynamics of the input SSMs.
First, given an input action video, we extract either low-level features like silhouettes in a frame-by-frame manner, or middle-level features like HOG3D from the partitioned video blocks, or some tracked feature points. Second, we transform the feature vector in each frame into an SSM. From the sequence of SSMs we then construct a symmetric and unique 3D structure, which we refer to as the Joint Self-Similarity Volume. This volume carries characteristic information about action dynamics. However, in order to exploit it more efficiently, and handle its large dimension, it is decomposed into three compact and discriminative vectors, two of which are identical (due to symmetry). These descriptor vectors characterize the internal dynamics of an action. Finally, the vectors are used for measuring distances for final nearest neighbor classification.

Figure 1: The flow of our action recognition framework. First, given an input action video, we extract either low-level features like silhouettes in a frame-by-frame manner, or middle-level features like HOG3D from the partitioned video blocks, or some tracked feature points. Second, we transform the feature vector in each frame into an SSM. From the sequence of SSMs we then construct a symmetric and unique 3D structure, which we refer to as the Joint Self-Similarity Volume. This volume carries characteristic information about action dynamics. However, in order to exploit it more efficiently, and handle its large dimension, it is decomposed into three compact and discriminative vectors, two of which are identical (due to symmetry). These descriptor vectors characterize the internal dynamics of an action. Finally, the vectors are used for measuring distances for final nearest neighbor classification.

Figure 2: Visualization of the symmetric Joint SSV. The middle figure shows its cut in three direction. The right figure shows the X-section of the volume.

4. Joint SSV construction

Our objective is to construct a 3D structure that can characterize the dynamics of the input vectors based on two SSM theorems. Spatiotemporal volume based analysis, in contrast to the bag-of-features (BoF), is preferable not only because local descriptors can be extracted, but also many local contextual information at spatiotemporal key-points can be preserved. Spatial and temporal saliency information embedded in the volume plays a key role in characterizing the volume. On the other hand, SSM is a representation that is already experimentally proven to be very discriminative for action recognition [13]. However, in [1, 5, 4, 13] each video is simply reduced to one single 2D SSM, and many useful and salient local structures might potentially be lost. For this reason, our aim is to embed the SSM representation into a characteristic 3D structure for better recognition.

Given an SSM per frame of an action sequence, one obtains directly a 3D volume of SSMs in the temporal dimension. This would yield a huge volume of information that captures the dynamics of the action. To tackle the dimensionality of the problem, we then use a rank-1 tensor decomposition to reduce it to a set of three compact vectors. This allows us to include as much information as needed into the 3D volume by fusing additional Joint SSMs before decomposition, without compromising classification accuracy as illustrated by experimental results later.

Suppose we have vectors $\Psi = \{V_1, V_2, ..., V_t\}$ with $V_i \in R^d$. These vectors can be regarded as some specific feature vectors varying over time $T$, say, extracted features from video sequence. Our objective here is to build a unique volume that simultaneously characterizes the dynamics of not only each element of $\Psi$ but also the relation amongst consecutive ones. Let $\Gamma: R^d \to R^{d \times d}$ be the operator that maps vector $v \in R^d$ to a self-similarity matrix $M^v$. According to Theorem 1, we have $\{M^v\}_{i=1...t}$, which uniquely models the internal dynamics for each individual vector. Based on the recurrent plot theory, the Laplacian operator is applied on the SSM sequence to fuse the consecutive SSMs. We define the gradient operator $\nabla_i$ on $\Psi$ as $\nabla_i \Psi = \frac{d\Psi}{dt} = V_i - V_{i-1}$. Since $\Gamma(\Psi) = \{\Gamma(V_i)\}_{i=1...t}$, we have $\nabla_i \Gamma(V_i) = \Gamma(V_i) - \Gamma(V_{i-1})$. Note that according to Theorem 2, we have $\Gamma(\nabla_i(V_i)) = \Gamma(V_i - V_{i-1}) = \nabla_i \Gamma(V_i)$, therefore, $\Gamma \nabla_i^2 (\Psi) = \nabla_i^2 \Gamma(\Psi)$, and we can further arrive at the following theorem.

**Theorem 3:** Given a random vector $\Psi$ and a self-similarity matrix operator $\Gamma: R^d \to R^{d \times d}$, it holds that $\Gamma \nabla_i^2 (\Psi) = \nabla_i^2 \Gamma(\Psi)$.

The self-similarity matrix operator $\Gamma$ and the second order Laplacian operator $\nabla_i^2$ are exchangeable in terms of the SSM sequence. Now we define the Joint Self-Similarity Volume based on Definition 2. Let $\circ$ be the element-wise multiplication operator between two matrices.

**Definition 3:** The Joint Self-Similarity Volume corresponding to a random vector $\Psi$ is built via a map $\Xi : R^{d \times T} \to R^{d \times d \times T}$ such that each element in $T$ dimension is defined by a matrix satisfying $\Xi_i |_{i=1...T} = \Gamma(V_i) \circ \nabla_i^2 \Gamma(\Psi_i)$. This generates a symmetric 3D volume.

5. Rank-1 tensor approximation

Numerous methods have been proposed in recent years for extracting discriminative descriptors from spatio-temporal volumes. Here we address this problem of extracting descriptors from Joint SSV from a different perspective. Since the Joint SSV is a characteristic, unique, and a symmetric 3D structure, we developed an iterative algorithm using the tensor theory to extract the most compact and optim
Figure 3: Rank-1 approximation \( \hat{A} = \lambda U^{(1)} \circ U^{(2)} \circ U^{(3)} \) for original Joint SSV \( \mathcal{A} \).

The Joint SSV can be considered as a 3-order tensor in multilinear algebra. The rank of tensor \( \mathcal{A} \) is defined as the minimum number of rank-1 tensors that sum up to \( \mathcal{A} \) [6], and a tensor is said to be rank-1 if it can be expressed as an outer product of a number of vectors. Although the truncation of the high-order SVD (HOSVD) of a given tensor may lead to a good rank-(\( R_1, R_2, ..., R_N \)) approximation, it is known that this will not necessarily generate the best possible (least-squares) approximation under the given \( n \)-mode rank constraints [15].

To obtain an optimal rank-1 approximation of Joint SSV, we propose an alternating least-squares (ALS) method by optimizing the components of the factorization of a given SSV in an iterative fashion similar to [15, 16]. Given a real \( N \)-th-order tensor \( \mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \), there exists a scalar \( \lambda \) and \( N \) unit-norm vectors \( U^{(1)}, U^{(2)}, \ldots, U^{(N)} \) such that a rank-1 tensor \( \mathcal{A} = \lambda U^{(1)} \circ U^{(2)} \circ \cdots \circ U^{(N)} \) minimizes the least-squares cost function

\[
J(\hat{A}) = \| \mathcal{A} - \hat{A} \|_2
\]

over the manifold of rank-1 tensors, which can be analyzed using the Lagrange multipliers and yields the following equations [6]:

\[
\mathcal{A} = U^{(1)T} \times_1 U^{(2)T} \times_2 U^{(N)T} = \lambda U^{(1)} \circ U^{(2)} \circ \cdots \circ U^{(N)},
\]

\[
\| U^{(n)} \| = 1.
\]

Specifically, our objective is to find a rank-1 approximation of Joint SSV such that there exists a scalar \( \lambda \) and three vectors \( U^{(1)}, U^{(2)} \) and \( U^{(3)} \) with objective function

\[
\min \sum_{i,j,k} (a_{ijk} - \lambda r_{i}^{(1)} \circ U_{j}^{(2)} \circ U_{k}^{(3)})^2,
\]

where \( a_{ijk} \) denotes the Joint SSV, a 3-order tensor, as shown in Fig.3. The \( \circ \) is the outer product operator for vector, \( i \) and \( j \) are spatial mode indices and \( i, j \in [1, I] \), \( I \) is the size of Joint SSM: while \( k \in [1, K] \), \( K \) is the frame number for this Joint SSV. Since each vector \( U^{(1)}, U^{(2)} \) and \( U^{(3)} \) is determined only up to a scaling factor, we have

\[
\| U^{(1)} \|_2 = \| U^{(2)} \|_2 = \| U^{(3)} \|_2 = 1.
\]

On the other hand, Joint SSV is symmetric in spatial dimension since its elements remain constant under any permutation of the indices \( i \) and \( j \), i.e. \( a_{ijk} = a_{jik} \), therefore

\[
U^{(1)} = U^{(2)} = U^{(3)}.
\]

For clarity of presentation, we denote \( U^{(1)}, U^{(2)} \) and \( U^{(3)} \) as \( \rho, \rho \) and \( \varepsilon \), and we will call them the primary vector \( \rho \), and secondary vector \( \varepsilon \), respectively. Under the constraint of Eq.(2), the Eq.(1) can be solved by the technique of Generalized Rayleigh Quotient (GRQ) in [30], and we adopt the alternating least squares algorithm (ALS) in this paper for the optimal SSV approximation. Although the rate of convergence of this method has not yet been analyzed in the literature, it is proved in [30] that linear convergence of ALS can be achieved in a neighborhood of the optimal solution by GRQ. We hence propose the Algorithm 1 for the optimal SSV approximation.

\begin{algorithm}
\caption{Joint SSV rank-1 approximation}
\begin{algorithmic}
\STATE \textbf{input} : A 3-order tensor Joint SSV \( \mathcal{A} \in \mathbb{R}^{I \times I \times K} \), where \( I \) is the spatial dimension of Joint SSM, and \( K \) is the temporal dimension of \( \mathcal{A} \).
\STATE \textbf{output}: Two vectors \( \rho \) and \( \varepsilon \) that minimize \( \| \mathcal{A} - \lambda \rho \circ \rho \circ \varepsilon \|_2 \), where \( \rho \in \mathbb{R}^I \), \( \varepsilon \in \mathbb{R}^K \), and \( \| \rho \|_2 = \| \rho \| = 1 \).
\STATE \textbf{Initialize} \( U^0 = [\rho(0), \varepsilon(0)]^T \).
\FOR {\( t \leftarrow 0 \) to \( N_{\text{maxiteration}} \)}
\STATE \( \hat{\rho}^{(t+1)} = \mathcal{A} \times_1 \rho^{(t)} \times_3 \varepsilon^{(t)} \);
\STATE \( \hat{\varepsilon}^{(t+1)} = \mathcal{A} \times_3 \rho^{(t)} \times_1 \rho^{(t)} \);
\STATE \( \rho^{(t+1)} = \hat{\rho}^{(t+1)} / \| \hat{\rho}^{(t+1)} \| \);
\STATE \( \varepsilon^{(t+1)} = \hat{\varepsilon}^{(t+1)} / \| \hat{\varepsilon}^{(t+1)} \| \);
\STATE \( \lambda^{(t+1)} = \mathcal{A} \times_1 \rho^{(t+1)} \times_3 \rho^{(t+1)} \times_3 \varepsilon^{(t+1)} \);
\ENDFOR
\end{algorithmic}
\end{algorithm}

In Algorithm 1, the \( \bar{x}_i \) for \( i = 1, 2, 3 \) denotes the multiplication between a tensor and a vector in mode-\( i \) of that tensor, whose result is also a tensor, namely,

\[
B = A \bar{x}_i \rho \iff (B)_{ijk} = \sum_{i=1}^{I} \mathcal{A}_{ijk} \rho_i.
\]

Starting with random initial values for \( \rho \) and \( \varepsilon \), the algorithm individually changes \( \rho \) (or \( \varepsilon \)) while fixing the other one and iteratively achieves the optimal approximation. The iteration stops when the difference between \( \mathcal{A} \) and \( \hat{A} \) arrives at a sufficiently small value, as illustrated in Fig.4. We observe that when set the iteration number be 5, it is experimentally sufficient to obtain a rank-1 tensor \( \hat{A} = \lambda \rho \circ \rho \circ \varepsilon \) that achieves the optimal approximation to the original SSV \( \mathcal{A} \).
6. similarity measure

We define the similarity measure between two initial input vectors using the decomposed descriptor vectors. Let $\Psi$ and $\Psi'$ be the two initial input vectors, whose corresponding decomposed vector pairs are $v = \{\rho, \rho', \varepsilon\}$ and $w = \{\rho', \rho', \varepsilon'\}$, respectively. According to Algorithm 1, the dimensions of $\varepsilon$ and $\varepsilon'$ are determined by the temporal dimension of the two tensors constructed from $\Psi_i$ and $\Psi_j$, respectively, while the dimension of $\rho$ and $\rho'$ are determined by corresponding spatial dimension of Joint SSMs. Therefore $\varepsilon$ and $\varepsilon'$ are not necessarily of equal size, and neither are $\rho$ and $\rho'$. We first normalize $\rho$ and $\rho'$ (as well as $\varepsilon$ and $\varepsilon'$) to zero mean and unit variance, and make $\rho$ and $\rho'$ (as well as $\varepsilon$ and $\varepsilon'$) of equal dimension. In our framework the similarity measure between $\Psi$ and $\Psi'$ is defined as

$$D(\Psi, \Psi') = \sum_{i=1}^{3} \max d(v_i, w_i)$$

where $d(v_i, w_i)$ is a function, which returns the vector containing the cross-correlation values between the two vectors $v_i$ and $w_i$.

7. Experiments

We evaluated our method on 3 well-known datasets: Weizmann, KTH, and the CMU MoCap dataset. Our goal was to evaluate the feasibility of our techniques on various datasets with different Joint SSV schemes.

7.1. Three schemes

**HOG3D-based Joint SSV (JSSV-hog3d)** We employ the dense representation as in [26], and use HOG3D descriptor [14] at densely distributed locations within a Region of Interest (ROI) centered around the actor, and partition the volume into overlapping blocks, and all blocks are then partitioned into small regular cells. Histograms of 3D gradient orientations, generated using dodecahedron based quantization [14] with 6 orientation bins, for cells within a block, are then computed, and concatenated to form a block descriptor.

We used the same configuration for defining ROIs but different block setup as in [26]. We used $16 \times 16 \times 2^7$ pixel blocks subdivided by $2 \times 2 \times 2$ cells, and computed the HOG3D descriptor for each block. Note that $\tau$ is generally from 1 to 5. Here we name all blocks within the same temporal location a slice, as shown in Fig.5. Slices overlap with each other between consecutive ones, yielding a redundant representation, which enhances the discriminative power [26]. Within each slice, all blocks are concatenated in row order into a block sequence. This sequence is a vector used for building the self-similarity matrix. Using all slices, we then construct a Joint SSV out of both SSMs and Joint SSMs using the procedure described in Section 4.

**Silhouette-based Joint SSV (JSSV-silh)** Silhouette feature of an action video frame has been extensively explored in the literature. We extract the contour from silhouette in each frame and transform the contour into time series using the method in [28], as shown in Fig.6. The time series are normalized to zero mean and unit variance before being fed into the framework as input vectors to generate the Joint SSV. Silhouettes can be easily extracted from static or uniform action background, but harder or even impractical for more challenging action sequences. For this reason, we merely test this scheme on Weizmann dataset, which provides well-extracted silhouette features. Fig.7 shows a sequence of generated SSMs for *Bend* action in Weizmann dataset, and Fig.8 shows the visual difference between four different actions.

**Tracked points based Joint SSV (JSSV-pos)** We evaluate our framework also on the CMU Mocap dataset containing tracked points at limbs for human actions. The dataset contains tracked points for different joints of human body for each action frame. We selected 13 key points, which we
Figure 7: A sequence of computed SSMs for frames selected from the the Bend action in Weizmann dataset. Note that all above SSMs have identical dimension. Both salient and delicate differences among silhouette contours can be revealed by SSMs.

Figure 8: SSM comparison among various action frames using silhouette feature. (Top) Selected frames from 4 actions Jack, Run, Wave2, and Side; (Bottom) The corresponding silhouette-based SSMs

contend can sufficiently define an action, and then concatenated them into a vector in the following order: left knee, right knee, left feet, right feet, heep, left elbow, right elbow, left shoulder, right shoulder, neck, head, left hand, and right hand, as shown in Fig.9. In this way, each frame of an action sequence can be initially represented by a vector of size 13 (Fig.9), which is used in the subsequent Joint SSV construction stage.

Table 1: Recognition rate comparison for various dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>Weizmann</th>
<th>KTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSSV-silh</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>JSSV-hog3d</td>
<td>100.0</td>
<td></td>
</tr>
</tbody>
</table>

Schindler [23] 100.0
Zhang [31] 97.8
J.Imran [13] 95.3
Niebles [20] 90.0
Liu [19] 89.3

Methods | CMU Mocap |
---------|-----------|
JSSV-pos | 93.1       |
Shen [25] | 92.0       |
J.Imran [13] | 90.5       |

7.2. Datasets and recognition rate

For all the results reported in this section, we performed the recognition using nearest neighbor classification and leave-one-out cross validation.

Weizmann dataset The Weizmann dataset consists of videos of 10 different actions performed by 9 actors. Each video clip contains one subject performing a single action. The 10 different action categories: walking, running, jumping, gallop sideways, bending, one-hand-waving, two-hands-waving, jumping in place, jumping jack, and skipping. Each of the clip lasts about 2 seconds at 25Hz with image frame size of 180 × 144.

We evaluated two schemes separately, namely the JSSV-silh and the JSSV-hog3d. For the former scheme, we used the provided well-extracted silhouettes in dataset to build input vectors for the whole framework, and we were able to achieve a recognition rate of 100%. For the latter one, we extract the ROI using the silhouettes and fitting a bounding box around each of them. To be consistent, all ROIs in our experiments are scaled and concatenated to form a 128 × 64 × t volume, where t is the frame number in sequence. We evaluated various block size setups (Fig.10) and observed that when τ = 3 (i.e. block size: 16 × 16 × 8), the JSSV-hog3d scheme yields the best recognition rate of 100%, as shown in Table 1.

KTH dataset The KTH dataset consists of 6 actions performed by 25 actors in four different scenarios. We follow the evaluation procedure in [26] but use slightly different settings for block size. We extract the ROIs using the bounding boxes provided by [18], and evaluate the JSSV-hog3d scheme on this dataset under various block size configurations, as shown in Fig.10. When τ is small, the block depth is small, making the final decomposed vectors undis-
Figure 9: Build SSMs for Run and Kick action in CMU Mocap dataset by stacking the 13 tracked key limb positions. The central figure in the first row shows the tracked point position and their indices for forming the initial vector. All SSMs are of the dimension $13 \times 13$ and are generated from the vector by concatenating the tracked limb points.

criminating for classification. But as $\tau$ grows, the recognition rate grows accordingly. This also agrees with our intuition that larger blocks contain more cells, and capture more stable gradient information comparing with the smaller ones. But as the block size becomes larger, more redundant information is introduced, leading a reduced recognition rate. Especially, our best recognition rate 100\% is achieved when $\tau = 4$. This outperforms both the result in [26] (92.4\%), which has similar experimental configuration to us, and the state-of-the-art in [10] (94.5\%).

**CMU Mocap dataset** We used motion capture data from CMU dataset to evaluate the performance of our framework. A total of 164 sequences corresponding to 12 actions (walkturn, golf, fjump, flystroke, jjack, jump, carwheels, drink, kick, walk, bend, run) under 6 cameras are used. We adopted the JSSV-pos scheme for this dataset, and are able to achieve a recognition rate of $\{89.0, 94.2, 95.7, 91.3, 93.3, 95.2\}$% for the 6 cameras, respectively, as shown in Fig.11. Our overall recognition rate averaged across all the 6 cameras is 93.1\%, which outperforms the method in [25] (92.0\%), but [25] adopted different evaluation setting using only 5 actions, each performed by 3 actors. Note that our approach also outperforms method in [13] (90.5\%), which also explored several SSM-based approaches.

8. Conclusion

In this paper, we study the application of Joint Self-Similarity Volume for action recognition in video sequences. A new optimized rank-1 tensor approximation algorithm is proposed for dimensionality reduction, which can largely preserves the salient characteristics for scene dynamics. A significant saving in both memory and computation complexity can be achieved since only a collection of rank-1 tensors is adopted as the reference database. The algorithm also allows one to recognize actions without explicitly aligning the videos in temporal dimension. Due to the fact that the proposed formulation is not dependent on the low-level features extracted from the sequence, we can apply this framework using any type low-level feature vector, including feature vectors that are view-invariant.

Figure 11: Recognition rate for Mocap dataset using JSSV-pos scheme. (Top) The recognition rates for 12 actions under 6 camera views; (Bottom) The confusion matrix averaged over 6 cameras.

References


